Modeling Web Services with URML

Adrian Giurca, Sergey Lukichev, Gerd Wagner

BTU Cottbus, Germany

June 11, 2006. SBPM Workshop, Budva, Montenegro.

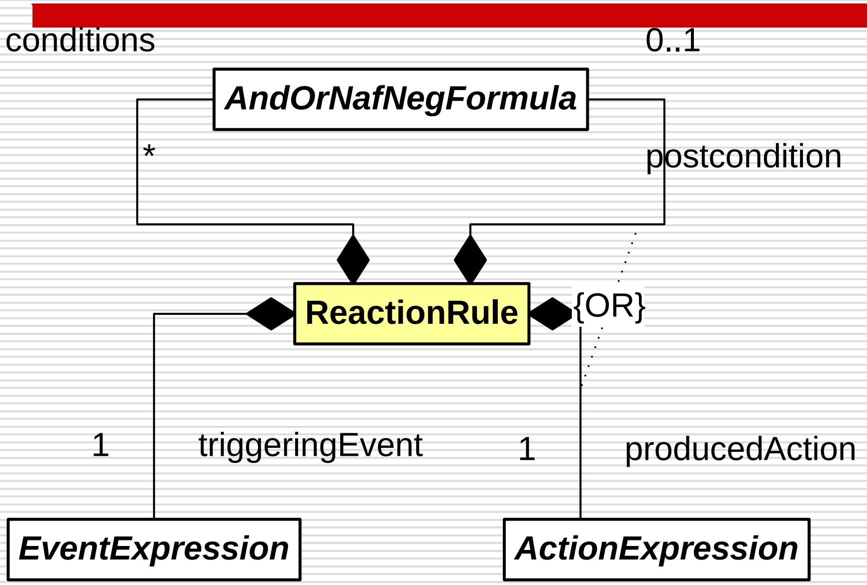
Outline

- Idea of Web Service modeling using Reaction Rules
- Reaction Rules - advantages
- A UML-Based Rule Modeling Language: motivation, examples
- R2ML – Rules Interchange Format
- Conclusions and open issues

Idea

- Apply rule modeling techniques, developed in REWERSE Working Group I1 for designing web services;
- Describe behavior of Web Services by means of Reaction Rules;
- Use UML-Based Rule Modeling Language for modeling web services.

Reaction (ECA) Rules



A *reaction rule:*

□ is a statement of programming logic that specifies the execution of one or more actions in the case of a triggering event occurrence and if rule conditions are satisfied;

□ has an operational semantics.

□ *Triggering event* which is either atomic or composite;

□ *Conditions* are represented as a collection of quantifier free logical formulas;

□ *Produced Action* represents the state change of the system;

□ *Postcondition* specifies a state change in a declarative manner.

Advantages of Using Rules

- □ Business requirements are often captured in the form of rules in a natural language, formulated by business people;
- □ Reaction rules are easier to maintain and integrate with other kinds of rules, used in business applications (integrity rules, which specify constraints the data must fulfill, derivation rules, which explain how a model element can be derived);
- □ The topic of rules validation and verification is well-studied;
- □ Reaction rules emphasis on events gives a flexible way to specify control flow and integrates events/actions from the real life;
- □ Declarative Programming versus Procedural Programming.

Modeling Scenario

- A modeler describes a Web services based on a vocabulary and rules, using modeling language (URML), supported by a modeling tool (Strelka);
- Consider 2 deployment options:
  - WSDL and Reaction Rules set are input parameters for the Web Service compiler, target platform is EJB (servlets).
  - Reaction Rules are mapped into execution language of a Reaction Rule engine (for example, RuleCore).

Rule Modeling: Idea and Motivation

- The modeling language should allow visual rule expressions, which can be understood by domain experts without extensive technical training;
- Modeling language is based on UML, which makes it usable for software engineers;
- Accommodate main concepts of UML, OCL and Semantic Web languages like SWRL.

Rule Modeling: Visual Rule Modeling Approach

- UML class diagram can be used for vocabulary definition;
- Rules are built on top of vocabularies;
- Visual rule modeling approach is implemented as an extension of UML metamodel with additional rule concepts;
- A Web service can be specified on the basis of a business vocabulary, including a business event model, and of a rule-based behavior model.

URML- A UML-Based Rule Modeling Language

- Extends UML metamodel to support derivation rules, production rules, and reaction rules;
- Provides visual notation for rules modeling;
- Forward compatibility with R2ML (R2ML serialization support by the modeling tool).

R2ML – The REWERSE I1 Rule Markup Language: Scope and Advantages

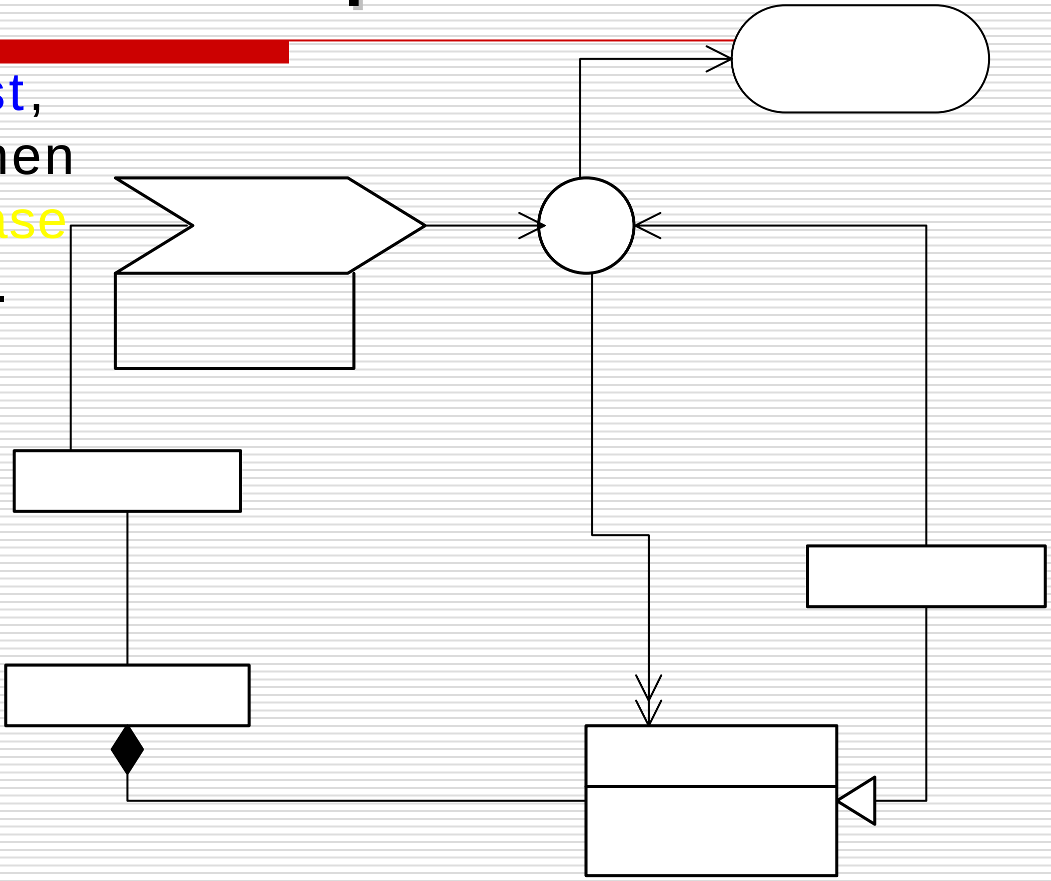
- □ An XML based rule language;
- □ Support for: Integrity Rules, Derivation Rules, Production Rules and Reaction rules;
- □ Integrate functional languages (such as OCL) with Datalog languages (such as SWRL);
- □ Serialization and interchange of rules by specific software tools (PIM level interchange format between different PSMs);
- □ Integrating rule reasoning with actual server side technologies;
- □ Deploying, publishing and communicating rules in a network.

R2ML: Design principles

- Modeled using MDA;
- Rule concepts are defined with the help of MOF/UML;
- Required to accommodate:
  - Web naming concepts, such as URIs and XML namespaces;
  - The ontological distinction between objects and data values;
  - The datatype concepts of RDF and user-defined datatypes;
- Actions (following OMG PRR submission);
- Events;
- EBNF abstract syntax;
- XML based concrete syntax validated by an XML Schema;
- Allows different semantics for rules.

Reaction Rule Example

On customer book request, if the book is available, then approve order and decrease amount of books in stock.



Visual notation:

- □Rule is represented as a circle
- □Triggering event is represented as a UML signal symbol
- □Precondition is an arrow from a classifier to a rule
- □Postcondition is a double head arrow
- □Action is a UML activity symbol

Events Model

- Composite events:
  - has a property `timeWindow`, which represents the duration of the corresponding event observation.
- Atomic Events:
  - has no duration;
  - refers to a classifier;
  - is composed from an ordered list of terms as arguments.
  - distinguishes between two main classes of atomic events: `MessageEventExpression` and `TimeEventExpression`.
  - A category of message event is a SOAP message event

Message Event Expression

- Has a property sender;
- A sender may be HTTP_REFERER;
- A category of message event is a SOAP message event:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:Body>
  <r2ml:SOAP-RPC-RequestMsgEvtExpr
            xmlns:r2ml="http://www.rewerse.net/I1/R2ML“
            r2ml:sender="eshop.com"
            r2ml:startTime="2006-03-21T09:00:00"
            r2ml:duration="P0Y0M0DT0H0M0S"
            r2ml:eventTypeID="productOrder">
     <r2ml:arguments>
            <r2ml:ObjectVariable r2ml:name="car" r2ml:classID="srv:Car"/>
     </r2ml:arguments>
 </r2ml:SOAP-RPC-RequestMsgEvtExpr>
</env:Body>
</env:Envelope>
```

Actions

- Following the OMG Production Rule Representation submission, an action is either
  - an InvokeActionExpression
  - an AssignActionExpression
  - a CreateActionExpression
  - a DeleteActionExpression
- The R2ML provides also message actions in the form of a concrete `SOAPMessageEventExpr`
- A SOAP message can be a triggering event for some rule and an action for the other: reaction rules can be chained

Strelka - Visual Rule Modeling Tool

- Fujaba (From UML to Java And Back Again) – Open source UML case tool
- Standard functionality can be extended by means of plug-ins
- Supports Java code generation for models
- Strelka – rule modeling tool, implemented as a Fujaba plug-in, supports URML and serializes rule models into R2ML

Conclusions and Open Issues

- UML-based rule modeling language can be used for describing behavior of web services
- R2ML as a rule markup framework;
- Open Issues:
  - Suitability of R2ML for expressing control-flow patterns;
  - The issue of implicit WSDL definition in URML has to be investigated (binding components);
  - Future integration of URML with web service standards (WSDL, SOAP).

For more info

Specifications, manual, use cases, downloads:

http://www.rewerse.net/I1